Objectives

- Understand the ARM v8-M architecture and its security features
- Learn about ARMv8-M Memory Protection mechanism enhancement
- Configuring the Security Attribution unit
- How to manage Security access faults
- How to build and debug a secure and non-secure software

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - One Online Linux PC per trainee for the practical activities.
    - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Prerequisites

- Programming skills: Some programming experience, particularly in C
- Basic knowledge of ARM Cortex-M implementations
- Basic understanding of Security Algorithms and Secure coding

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.

- Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

First Day

ARM Architecture and Security

- Overview of ARM TrustZone technology
- TrustZone Architecture
  - Overview of the TrustZone architecture
  - TrustZone-enabled processors and their features
  - Secure world and non-secure world
- TrustZone security
  - Overview of TrustZone security model
  - TrustZone-enabled Cortex-M
- Secure Software Design Considerations

ARMv8-M Memory Protection

- Memory types
- Access order
- Memory barriers, self-modifying code
- Memory protection overview, ARM v8 PMSA
- Cortex-M33 MPU and bus faults
- Region overview, memory type and access control
- Setting up the MPU

Exercise: *Use the MPU to protect an area of memory against unintended access*

Cortex-M TrustZone

- TrustZone-enabled Cortex-M processors and their features
- Security states
- Register banking between security states
- Stacks and security states
- Security Extension and exceptions
- Secure and Non-Secure states interactions
- Exceptions and the Security Extension
  - Handling Secure Exceptions
  - Handling Non-Secure Exceptions while in the Secure state
  - Returning from a Non-Secure exception to the Secure state
- The Security Attribution Unit (SAU)
- The Implementation Defined Attribution Unit (IDAU)
- Debugging TrustZone-enabled Cortex-M processors

Exercise: *Implementing a minimal secure monitor*

Exercise: *Programming and Debugging a TrustZone application example*

Renseignements pratiques

Inquiry : 1 day